R U :-) or :-(? Character- vs. Word-Gram Feature Selection for Sentiment Classification of OSN Corpora

Ben Blamey and Tom Crick and Giles Oatley

Abstract Binary sentiment classification, or sentiment analysis, is the task of computing the sentiment of a document, i.e. whether it contains broadly positive or negative opinions. The topic is well-studied, and the intuitive approach of using words as classification features is the basis of most techniques documented in the literature. The alternative character n-gram language model has been applied successfully to a range of NLP tasks, but its effectiveness at sentiment classification seems to be under-investigated, and results are mixed. We present an investigation of the application of the character n-gram model to text classification of corpora from online social networks, the first such documented study, where text is known to be rich in so-called unnatural language, also introducing a novel corpus of Facebook photo comments. Despite hoping that the flexibility of the character n-gram approach would be well-suited to unnatural language phenomenon, we find little improvement over the baseline algorithms employing the word n-gram language model.

1 Introduction

As part of our wider work on developing methods for the selection of important content from a user's social media footprint, we required techniques for sentiment analysis that would be effective for text from online social networks (OSNs). The n-gram model of language used by sentiment analysis can be formulated in two ways: either as a sequence of words (or overlapping sequences of n consecutive words), or more rarely, as a set of all overlapping n-character strings, without consideration of individual words.

Ben Blamey, Tom Crick, Giles Oatley
Cardiff Metropolitan University, Western Avenue, Cardiff, CF5 2YB, United Kingdom, e-mail: {beblamey,tcrick,goatley}@cardiffmet.ac.uk

M. Bramer and M. Petridis (eds.), *Research and Development in Intelligent Systems XXIX*,
DOI 10.1007/978-1-4471-4739-8_16,

Our hypothesis is that the character n-gram model will be intrinsically well-suited to the 'unnatural' language common to OSN corpora, and will achieve higher accuracy in the binary sentiment classification task.

For our study, we gathered 3 corpora: movie reviews, Tweets, and a new corpus of Facebook photo comments. We ran traditional word-based classification alongside character n-gram based classification. The aim was to see whether the character n-gram model offers improved accuracy on OSN corpora, with the movie review corpus serving as a non-OSN control.

2 Background & Related Work

Text found in social media is rich in 'unnatural' language phenomena, defined as "informal expressions, variations, spelling errors ... irregular proper nouns, emoticons, unknown words" [1]. Existing NLP tools are known to struggle with such language, Ritter et al. have "demonstrated that existing tools for POS tagging, Chunking and Named Entity Recognition perform quite poorly when applied to Tweets" [2, pp. 1532], Brody and Diakopoulos "showed that [lengthening words] is a common phenomenon in Twitter" [3, pp. 569], presenting a problem for lexicon-based approaches. These investigations both employed some form of inexact word matching to overcome the difficulties of unnatural language, we wondered whether the flexibility of the character n-gram language model would make it more appropriate than the word-based language model for sentiment analysis of OSN text.

The character n-gram model is arguably as old as computer science itself [4]. It has been proven successful for tasks within NLP; such as information extraction [5], Chinese word segmentation [6], author attribution [7], language identification [8], and other text-classification tasks [9, 10, 11], and outside it (e.g. compression, cryptography). The word-based model has a number of disadvantages: Peng et al. cite drawbacks of the standard text classification methodology including "language dependence" and "language-specific knowledge", and notes that "In many Asian languages ... identifying words from character sequences is hard" [12, pp. 110].

Research has found character n-grams to perform better than word n-grams: Peng et al. apply the character n-gram model to a number of NLP tasks, including text genre classification observing "state of the art or better performance in each case". [12, pp. 116]. NLP expert Mike Carpenter, has stated that he favours character n-gram models [13].

There are just a few examples of sentiment analysis employing the character n-gram model: Rybina [14] did binary sentiment classification on a selection of German web corpora, mostly product reviews, and finds that character n-grams consistently outperforms word n-grams by 4% on F1 score. This is an extremely interesting result, and our desire to repeat her findings were a key motivation for this work, but some details are unclear: the classifier that was used is closed-source, and it isn't obvious what method was used to label the data. Nevertheless, such a result demands further explanation. Other studies have more mixed findings; Ye et

al. [11] classified travel reviews using both character and word n-gram models with LingPipe, and found that neither was consistently better.

Much work has studied sentiment analysis of OSN corpora, especially Twitter, using the word n-gram model. Go et. al [15], in the first sentiment analysis study of the micro-blogging platform, achieved accuracy of around 80%. Pak and Paroubek [16] experimented with exclusion of word n-grams; based on saliency and entropy thresholds, but neither the thresholds themselves nor improvement over original accuracy are quoted. At the state of the art, Bespalov et al. [17] overcome the high-dimensionality of the word n-gram feature-set using a multi-layer perceptron to map words and n-grams into lower-order vector spaces, while retaining meaning. The approach achieves accuracy of more than 90%.

Sentiment analysis studies of Facebook are comparatively rare, in one such study Kramer computed the positivity of status messages using LIWC[1] [18] to create an index of "Gross National Happiness" [19]. To our knowledge, there have been no documented studies of sentiment analysis applying the character n-gram model to online social network text, and none looking at Facebook photo comments using either language model.

3 Methods

An emoticon is a sequence of characters with the appearance of a facial expression. For Tweets and Facebook photo comments, we follow the approach of Read [20] of using the presence of emoticons as labels of sentiment (thereby allowing unsupervised machine learning), and them removing them from the text for classification (similarly, the standard approach is to use 'star-ratings' with movie reviews). We distinguish the emoticons from instances where the text resembles something else, such as <3 (a heart). The importance of unnatural language is exemplified by one Facebook photo comment, reading simply: "<3!". A comprehensive list of Western emoticons was compiled from Wikipedia[2]. Around 20,000 Tweets were gathered from the Twitter Search API, using positive and negative emoticons as search terms. With Facebook photo comments, the percentage of comments that contained emoticons was low, (negative emoticons were found in less than 1% of the corpus), so it was necessary to collect a large amount of data. We managed to obtain over 1 million unique Facebook photo comments via the Facebook API.

URLs and Twitter 'mentions' and hashtags were replaced with respective single characters, so their meaning is captured in both word and character n-gram models. Only the documents with exclusively happy {:) :D :-) =) :) :-D :o) =D} or sad {:(:-(:'(: (} emoticons only were selected, and emoticons were chosen that accurately mean 'happy' and 'sad' - documents with :P ('sticking out tongue') and similar emoticons were excluded, because they tend to be used for jokes and

[1] http://www.liwc.net/

[2] http://en.wikipedia.org/wiki/List_of_emoticons

insults - which might confuse the classifiers. Also excluded were 'winks', e.g. ';-)', incase they were used to indicate flirtatious remarks, which again may disrupt classification, for example: "Nice slippers, hon! (Brother's not bad either... ;-) tee hee!) x". Emoticons are replaced with a period, because they seem to be used as end-of-sentence punctuation, indeed they are commonly suffixed directly onto other words, so it is best to search for them as a substring (technically an application of the character-based model). This yielded labelled corpora of 7000 Tweets and 7000 Facebook photo comments, alongside 1386 movie reviews used in previous studies [21].

For word-based classification, further processing is necessary. Elongated words are squashed to a maximum of 3 repetitions, e.g. '<3<3<3<3<3' becomes '<3<3<3'. Go et al. [15] handle single repeated characters only, shortening to two reptitions. By shortening to three, we preserved the elongated variations as separate features, as word length is known to indicate sentiment strength [3]. We followed the approach of Das and Chen [22] to handle negation (as in "I am not happy") by labelling words in a negative context, yielding a small improvement to classification, consistent with other studies.

For the text classification itself, we used 4 feature-sets: word unigrams, bigrams, and the union of both, and the union of the sets of character n-grams where $n \in \{1,..,8\}$. Note that we do not trim low-frequency features, as it is generally discouraged except where necessary for performance. Three standard classifiers were used in our experiment:

- Naive Bayes (based on feature frequency), with 'plus-one' smoothing.
- Maximum Entropy (i.e. 'loglinear discrimitive') of the Stanford Classifier[3], with default settings (a quadratic prior with $\sigma = 1$).
- SVM^{light} classifier[4] [23], with default settings.
- For character n-grams only, the LingPipe[5] [24] DynamicLMClassifier.

The classification accuracies are shown below in Table 1.

4 Conclusions

Our results look a lot like those of Ye et al. [11], with neither word- nor character-grams yielding consistently higher accuracy. Therefore, the findings contradict some existing studies: inconsistency with the results of Rybina [14] (a consistent 4% improvement with character n-grams). Her results are hard to explain – language is a slight possibility (her corpus was German).

Looking more closely at our data, we can see that character n-grams consistently beat word unigrams, which is understandable, as 8 characters will often be enough

[3] http://nlp.stanford.edu/software/classifier.shtml

[4] http://svmlight.joachims.org/

[5] http://alias-i.com/lingpipe/

to contain more than one word, and including word bigrams has often given better accuracy than unigrams alone.

Our hypothesis that character n-grams will be intrinsically well-suited to the 'unnatural' language common to OSN corpora was false: there doesn't seem to be a significant performance difference between the OSN and non-OSN corpora, for social network text and unnatural language – but the language of the social web is a pressing challenge for NLP; and as discussed, many of the existing tools struggle with it. The size of our corpus may have been an issue, to reap the full benefits of the character n-gram model more training data might be needed - LingPipe is designed to scale character n-gram data to the order of gigabytes [25].

Putting the issue of unnatural language aside, proponents of character n-gram models have a point: studies (including this one) have repeatedly shown that the character n-gram can perform as well as simple word n-gram models – whilst being considerably simpler to implement, especially when tokenization is hard, such as in Asian languages. There is no one 'right' way to do tokenization, negation, word squashing, stemming, and precise details are often thought too tedious for publication. Experiments involving word-grams can sometimes be difficult to repeat perfectly for these reasons, and greater use of character n-gram based algorithms would eliminate these inconsistencies between work. Our tendency to automatically adopt the word-based model may suggest some degree of human-centric bias in our research thinking, or perhaps too strong a focus on English and other Western languages, within sentiment analysis research.

Table 1 3-fold cross-validated accuracies. The best performing configuration of feature-set and classifier for each corpus is shown in bold.

Corpus	Features	Bayes	MaxEnt	SVM	LingPipe
IMDB Movie Reviews	Word Unigrams	80.5%	82.7%	82.8%	
	Word Bigrams	80.5%	79.4%	76.7%	
	Word Unigrams+Bigrams	81.4%	83.5%	82.2%	
	Character n-grams	81.9%	**84.6%**	82.6%	75.9%
Tweets	Word Unigrams	88.7%	91.2%	88.8%	
	Word Bigrams	89.5%	91.4%	90.5%	
	Word Unigrams+Bigrams	90.6%	91.6%	90.8%	
	Character n-grams	90.8%	91.9%	90.6%	**92.0%**
Facebook Photo Comments	Word Unigrams	80.2%	79.8%	78.6%	
	Word Bigrams	75.8%	75.8%	72.5%	
	Word Unigrams+Bigrams	**80.5%**	80.0%	78.3%	
	Character n-grams	80.4%	80.1%	80.0%	75.9%

References

1. M. Hagiwara. Unnatural Language Processing Contest 2nd will be held at NLP2011 (2010). URL http://bit.ly/dGvUnR
2. A. Ritter, S. Clark, Mausam, O. Etzioni, in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (ACL, Edinburgh, Scotland, UK., 2011), pp. 1524–1534
3. S. Brody, N. Diakopoulos, in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (ACL, Edinburgh, Scotland, UK., 2011), pp. 562–570
4. C. Shannon, Bell System Technical Journal (27), 379 (1948)
5. D. Klein, J. Smarr, H. Nguyen, C. Manning, in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2003), CONLL '03, pp. 180–183. DOI 10.3115/1119176.1119204
6. N. Xue, Computational Linguistics and Chinese Language Processing **8**, 29 (2003)
7. F. Peng, D. Schuurmans, S. Wang, V. Keselj, in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2003), EACL '03, pp. 267–274. DOI 10.3115/1067807.1067843
8. W.B. Cavnar, J.M. Trenkle, in *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* (1994), pp. 161–175
9. S. Raaijmakers, W. Kraaij, in *ICWSM* (2008)
10. R. Pon, A. Cárdenas, D. Buttler, T. Critchlow, in *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on* (2007), pp. 354–361. DOI 10.1109/CIDM.2007.368896. URL http://dx.doi.org/10.1109/CIDM.2007.368896
11. Q. Ye, Z. Zhang, R. Law, Expert Syst. Appl. **36**(3), 6527 (2009). DOI 10.1016/j.eswa.2008.07.035
12. F. Peng, D. Schuurmans, S. Wang, in *Proc. of HLT-NAACL 03* (2003), pp. 110–117
13. B. Carpenter. Yahoo group message discussion (2010). URL http://tech.dir.groups.yahoo.com/group/LingPipe/message/917
14. K. Rybina, Sentiment analysis of contexts around query terms in documents. Master's thesis (2012)
15. A. Go, R. Bhayani, L. Huang, Processing **150**(12), 1 (2009)
16. A. Pak, P. Paroubek, in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)* (ELRA, Valletta, Malta, 2010)
17. D. Bespalov, B. Bai, Y. Qi, A. Shokoufandeh, in *Proceedings of the 20th ACM international conference on Information and knowledge management* (ACM, New York, NY, USA, 2011), CIKM '11, pp. 375–382
18. F.M..B.R. Pennebaker, J.W. Linguistic inquiry and word count: Liwc2001 (2001)
19. A.D.I. Kramer. Facebook gross national happiness application (2010). URL http://www.facebook.com/gnh/
20. J. Read, Proceedings of the ACL Student Research Workshop on ACL 05 **43**(June), 43 (2005)
21. B. Pang, L. Lee, S. Vaithyanathan, in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, 2002), pp. 79–86
22. S. Das, M. Chen, in *Asia Pacific Finance Assc. Annual Conf. (APFA)* (2001)
23. T. Joachims, *Making large-scale SVM learning practical* (MIT press, 1999)
24. Alias-i. Lingpipe 4.1.0 (2008). URL http://alias-i.com/lingpipe
25. B. Carpenter, in *Proceedings of the Workshop on Software* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2005), Software '05, pp. 86–99